



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
DIRECCIÓN GENERAL DE LA ESCUELA NACIONAL COLEGIO  
DE CIENCIAS Y HUMANIDADES  
SECRETARÍA ACADÉMICA



DEPARTAMENTO DE OPCIONES TÉCNICAS

**PROGRAMA DE ESTUDIOS TÉCNICOS ESPECIALIZADOS EN  
SISTEMAS COMPUTACIONALES, DESARROLLO DE SOFTWARE**

Fecha de actualización:  
**10 de marzo al 30 de abril de 2015**

Fecha de aprobación por el Consejo Académico de Opciones Técnicas:  
**18 de junio de 2015**

Fecha de aprobación por el Consejo Técnico:  
**29 de noviembre de 2016**

Fecha de aprobación por el Consejo Académico del Bachillerato:  
**28 de junio de 2017**

Revisión y actualización con apego a los lineamientos por:

Lic. Diego González Sánchez  
Prof. Pedro Ruiz Tapia





## **PROGRAMA DE ESTUDIOS TÉCNICOS ESPECIALIZADOS EN SISTEMAS COMPUTACIONALES, DESARROLLO DE SOFTWARE**

### **I. PRESENTACIÓN**

La oferta de formación técnica complementaria en la Escuela Nacional Colegio de Ciencias y Humanidades (CCH), casi desde su inicio como parte del Plan de Estudios original, ha contribuido a la integración de los conocimientos y habilidades que se imparten a los alumnos como formación integral del mismo, a través de las asignaturas del Plan de Estudios, mediante el ofrecimiento que se les hace de cursar alguna de las especialidades técnicas que se ponen a su alcance en cualquiera de los cinco planteles.

Uno de los fines fundamentales del Departamento de Opciones Técnicas es ofrecer una formación para el trabajo a través de un adiestramiento teórico-práctico a nivel técnico que capacitará al alumno para incorporarse productivamente en el escenario laboral, todo esto manteniendo los principios fundamentales del CCH, mismos que orientan su filosofía y se traducen en los postulados pedagógicos que dirigen su tarea educativa en el nivel de la educación media superior de la UNAM, a saber: aprender a aprender, aprender a hacer y aprender a ser.

Actualmente, se considera a los Estudios Técnicos Especializados (ETE) como programas que buscan proporcionar a los alumnos una formación profesional para el trabajo, así como una orientación propedéutica y vocacional. Además, al cursar una especialidad técnica el educando adquiere conocimientos y habilidades de un campo laboral diferenciado, tiene la posibilidad de vincular la teoría con la práctica y fortalecer los conocimientos que adquiere en las materias del Plan de Estudios, apoyando así su formación integral.

Hoy en día se ofrecen 15 programas de Estudios Técnicos Especializados, y cuatro más se encuentran en fase piloto, la mayoría de ellos comprenden cursos-talleres específicos de la especialidad junto con la realización de actividades prácticas en centros laborales. Las actividades prácticas se les concibe como un aprendizaje basado en evidencias, una forma pedagógico-didáctico mediante la cual el alumno puede consolidar lo que aprende y sabe, confrontándolo con la realidad laboral, en la que además aprende nuevas cosas. Los ETE que no contemplan actividades prácticas se conforman como programas teórico-prácticos en los que se hace énfasis en un enfoque de aplicación de conocimientos y desarrollo de habilidades.

Por las características propias de los ETE se buscó que el diseño de los programas académicos estuviera apegado a la normatividad vigente, sin dejar a un lado la importancia del fortalecimiento de las actuales Tecnologías de la Información y la Comunicación (TIC) en el proceso enseñanza-aprendizaje.



## II. ANTECEDENTES

El modelo educativo del Colegio de Ciencias y Humanidades considera, desde su creación en 1971, como un ámbito de sustancial importancia, ofrecer opciones de estudios técnicos para sus alumnos, como lo establecen las Reglas y criterios de aplicación del plan de estudios de la Unidad Académica del Ciclo de Bachillerato del Colegio de Ciencias y Humanidades publicados en la *Gaceta UNAM* del 1 de febrero de 1971. Las especialidades técnicas se han venido impartiendo en el Colegio como estudios de carácter optativo para que los estudiantes sean capacitados en distintas vertientes, una de ellas es la “incorporación más rápida al mercado de trabajo, en salidas laterales, que son indispensables para un país moderno”<sup>1</sup> y que en la actualidad se presentan como Estudios Técnicos Especializados que formarán Técnicos auxiliares a nivel bachillerato.

Inicialmente, el Departamento de Opciones Técnicas realizó un modelo que intentaba atender a un gran número de alumnos, esto significaba enviar al mercado de trabajo grandes cantidades de mano de obra que con el tiempo saturaría la demanda de cuadros calificados; después vino un segundo modelo que, por su conformación, no posibilitaba la atención masiva de alumnos por lo que el número de egresados de las especialidades técnicas no representarían un impacto en el mercado de trabajo. De estos dos modelos surge, en el año de 1978, una propuesta que toma como punto de partida la investigación en el medio laboral con el objeto de detectar necesidades sociales e incorporarlas a la práctica laboral, misma que se ocupa hasta la actualidad.

En el año de 1979, el Consejo Académico de Opciones Técnicas aprobó el programa de la Especialidad Técnica de Programación: Lenguaje Cobol, la cual comienza a impartirse en el semestre 1980-1 hasta el semestre 1987-1. En este semestre comienza la aplicación del Plan Piloto de Computación para la Administración por considerarse que este programa estaba más acorde con el desarrollo reciente de la computación y con las necesidades de formación de recursos humanos detectados en el área.

Durante 1989 se aplicó un programa de Computación en acuerdo con del Director de la Unidad Académica del Ciclo del Bachillerato del Colegio de Ciencias y Humanidades. Este programa se estructuró una vez terminado y analizado el Plan Piloto iniciado en 1987 por la Profa. Evelina Chiu Ley. En el periodo de 1994 a 1997 se implementó la Opción Técnica en Computación.

A partir de 1998 se aplicó un nuevo programa para la Opción Técnica de Computación, la cual cambió de nombre a Sistemas Computacionales. Este programa es el resultado de una revisión hecha por el conjunto de profesores que impartían la opción,

---

<sup>1</sup> *Gaceta UNAM*, 1 de febrero de 1971 p. 2



en la cual se actualizaron y añadieron temas con el afán de ponerse al día y adecuarse al nuevo Plan de Estudios del Colegio y así continuar con la labor que se venía realizando.

En este momento, los avances tecnológicos, y en especial en el área de computación, son muy rápidos, lo que ha provocado la necesidad de adecuar el programa varias veces, siendo éste, otro momento para ello. Los Estudios Técnicos Especializados en Sistemas Computacionales, Desarrollo de Software estarán compuestos de dos fases, incorporando el análisis y diseño de sistemas, además de una introducción a los lenguajes visuales. Se mantiene el lenguaje de Programación “C”, con algunos reajustes a su secuencia y se elimina la parte referente a paqueterías (software de aplicación), ya que este ETE está enfocado al desarrollo de sistemas por medio de un lenguaje de programación.

Tomando en cuenta que en el Plan de Estudios Actualizado (PEA) de 1996 se señala que cada Opción Técnica tiene un Programa de Estudios, que puede ser modificado a partir de las reglas marcadas por el Departamento de Opciones Técnicas, así como las indicaciones del Reglamento General de Estudios Universitarios y de los Lineamientos de los Estudios Técnicos Especializados publicados en 2014 y 2015 respectivamente, el Departamento de Opciones Técnicas se da a la tarea de realizar una revisión y adecuación de los programas en la que se incluyen tópicos importantes como el carácter optativo y teórico-práctico de dichos estudios, la transversalidad de los aprendizajes frente a asignaturas del área curricular, la forma estratégica para su desarrollo, los requisitos de ingreso, permanencia y egreso, la duración, los créditos correspondientes y, finalmente el grado de Técnico especializado que obtiene respaldado con el diploma que se otorga por parte de la Institución.

### III. JUSTIFICACIÓN

#### *Social*

El campo laboral en materia de desarrollo de software es muy diverso; existe programación estructurada, orientada a eventos, a objetos, realidad y mundos virtuales y el uso de Herramientas CASE. Hoy en día, sociedad y empresa tienen la necesidad de un programador que brinde apoyo técnico y colabore en la interpretación de diagramas, identificando las condiciones de la empresa, ofreciendo alternativas de solución y diseñando software conforme a sus necesidades.

Con respecto a la sociedad, el técnico puede ser un emprendedor autosuficiente que desarrolle pequeñas aplicaciones para personas, instituciones o empresas que requieran de aplicaciones básicas y cuenten con pocos recursos para su implementación.



### *Laboral*

Los Estudios Técnicos Especializados en Sistemas Computacionales, Desarrollo de Software constituyen una opción que permite al alumno formarse en términos laborales y académicos para poder integrarse al trabajo a la par que estudia; además de que le ayuda a adquirir los conocimientos necesarios para decidir la carrera que estudiará a Nivel Superior, teniendo con esta Especialidad Técnica un complemento en su formación profesional; todo lo anterior a través del análisis y desarrollo de sistemas, así como de la implementación de un lenguaje de programación de alto nivel, además de una introducción a la programación orientada a objetos mediante un lenguaje visual.

Los ETE en Sistemas Computacionales, Desarrollo de Software se actualizan como una forma de mantener vigente el campo de estudio adecuado a la tecnología, los métodos y técnicas de programación. Además de ello, forma técnicos capaces de migrar aplicaciones entre los sistemas realizados con programación estructurada a sistemas visuales que pueden operar en sistemas operativos actuales de 32 o 64 bits que han dejado de lado o evitado la compatibilidad con la programación a 8 o 16 bits.

### *Institucional*

Nuestros alumnos deben desarrollar sus habilidades y destrezas en el planteamiento de soluciones a problemas académicos pero, también, en el ámbito general, por lo que este ETE cumple con su papel propedéutico contribuyendo en el desarrollo de habilidades y destrezas en los alumnos.

- a) Transversalidad o verticalidad de los aprendizajes del Estudio Técnico Especializado con otros ETE, así como con las asignaturas del área curricular

Este programa de ETE tiene una relación directa con las materias curriculares de Cibernética y Computación I y II, ya que ambas, además de apoyar los principios del Colegio, contribuyen en la formación integral de los alumnos que no sólo es básica sino también necesaria en la actualidad, debido a que es imprescindible contar con conocimientos de computación en el mundo de la programación estructurada. Los ETE, de manera complementaria, definen la programación estructurada como la base de las técnicas de programación actual, además de adentrar al alumno en el estilo de programación orientada a eventos.

Estos ETE tienen una relación interdisciplinaria con Mantenimiento de Sistemas de Microcómputo, porque mientras este último define la configuración óptima de los equipos adecuándola a las necesidades del usuario, Sistemas Computacionales, Desarrollo de Software diseña programas de cómputo compatibles con los equipos con los cuales se trabajará en el desarrollo



de aplicaciones; asimismo, puede relacionarse de manera indirecta con los ETE en Administración de Recursos Humanos, Contabilidad con Informática, Instalaciones Eléctricas en Casas y Edificios, Banco de Sangre y casi todas las especialidades técnicas para generar recursos y aplicaciones acordes con los requerimientos de éstas. En cuanto al contenido temático, se relaciona con ellas por el conocimiento de los sistemas operativos y las funciones elementales de los programas que residen en un equipo de cómputo.

#### b) Relación del Estudio Técnico Especializado con las licenciaturas

En el ámbito laboral podemos situar al Técnico egresado de los ETE en Sistemas Computacionales, Desarrollo de Software como un auxiliar de los Ingenieros de Sistemas, Licenciados en Informática y, en general, de empresas dedicadas al diseño de aplicaciones en el análisis y estudios de factibilidad para la implementación de un sistema automatizado, al análisis e interpretación de diagramas de flujo, así como llevar a cabo pruebas de funcionalidad de la aplicación desarrollada.

El carácter disciplinario de los contenidos y aprendizajes del Programa de Estudios Técnicos Especializados en Sistemas Computacionales, Desarrollo de Software promueve un conjunto de conocimientos, habilidades, actitudes y valores que permite al egresado integrarse posteriormente a alguna de las siguientes licenciaturas:

- Actuarial
- Administración
- Ciencias de la computación
- Ciencias genómicas
- Diseño industrial
- Informática
- Ingeniería Eléctrica y Electrónica
- Ingeniería en Computación
- Ingeniería en Telecomunicaciones
- Ingeniería Geomática
- Ingeniería Mecatrónica
- Matemáticas Aplicadas y Computación



#### **IV. ENFOQUE DE LOS ESTUDIOS TÉCNICOS ESPECIALIZADOS**

Con el uso de las herramientas computacionales en todos los campos del quehacer humano es indispensable que un bachiller adquiera los conocimientos que le permitan utilizar estas nuevas tecnologías en el manejo, análisis, procesamiento y automatización de la información, de la manera más adecuada y eficaz posible.

Es conveniente destacar que deberá mostrarse al alumno que el ETE capacita en la solución de problemas mediante el análisis y el diseño de un sistema que permita automatizarlo, valiéndose de una herramienta computacional para su implementación, sin que esta última sea la finalidad del curso, sino todo el proceso que le permitirá decidir cuál es la herramienta computacional adecuada y usarla con base en esta certeza.

Los contenidos deberán ser abordados con un nivel medio de complejidad, considerando que no existe ninguna preparación previa; pero conforme el curso avance, el grado de complejidad deberá también incrementarse hasta alcanzar casos que el alumno pueda encontrar en el campo laboral. Los tratamientos teóricos superfluos o demasiado extensos, deberán evitarse y ser preferentemente prácticos. Se cuidará sin embargo, que el estudiante entienda las relaciones entre los conceptos y la práctica, lo mismo que la estructuración de la programación.

Se recomienda que los problemas propuestos en clase primeramente se ejemplifiquen y resuelvan por el profesor y posteriormente el alumno plantee la solución a problemas similares, siempre con miras a enfrentarse a problemas de casos reales.

#### **V. PROPÓSITO GENERAL DEL PROGRAMA DE FORMACIÓN DEL TÉCNICO ESPECIALIZADO**

El propósito del ETE en Sistemas Computacionales, Desarrollo de Software tiene como propósito formar técnicos capacitados en desarrollar aplicaciones computacionales basadas en el análisis de problemas y diseño de sistemas, a través del uso del lenguaje de programación estructurada C o C++, que les permitirá separar y simplificar procesos a desarrollar donde se utilicen sentencias de control, funciones, estructuras de datos, gráficos y archivos de acuerdo con las necesidades de la sociedad que lo requiera. Además, el egresado podrá interpretar y documentar todo programa que este escrito, al contar con conocimientos de programación estructurada y de programación en lenguaje visual, lo cual permitirá realizar migración de aplicaciones en ambos lenguajes.



## VI. PERFIL DE EGRESO DEL TÉCNICO ESPECIALIZADO

El alumno egresado de esta opción estará capacitado para desarrollar sistemas computacionales que den solución a problemas que impliquen la automatización básica de procesos; dominará un lenguaje de programación que le permita implementar en la computadora los sistemas desarrollados, además de contar con las bases de la programación en lenguajes visuales.

Conocimientos	Actitudes	Habilidades	Valores
<ul style="list-style-type: none"><li>• Datos de entrada y salida para el planteamiento de su solución.</li><li>• Técnicas de programación estructurada y modular.</li><li>• Software para el uso del lenguaje de programación en la computadora.</li><li>• Configuración de software para lenguaje de programación.</li><li>• Lenguaje de programación para automatización de los sistemas en una computadora.</li><li>• Medios de almacenamiento para extraer y guardar datos utilizados en los programas.</li><li>• Desarrollo de aplicaciones en ambientes gráficos.</li><li>• Técnicas de ordenación y clasificación de datos.</li><li>• Diseño de sistemas de manejos de archivos.</li><li>• Lenguaje visual.</li><li>• Aplicaciones asociadas a eventos en un ambiente gráfico.</li></ul>	<ul style="list-style-type: none"><li>• Disposición para trabajar en equipo.</li><li>• Responsabilidad para el estudio.</li><li>• Responsabilidad para realizar las actividades prácticas.</li><li>• Interés por aprender por cuenta propia.</li><li>• Autocrítica.</li></ul>	<ul style="list-style-type: none"><li>• Realizar cálculos numéricos.</li><li>• Comunicar y transmitir mensajes.</li><li>• Desarrollar la escucha empática.</li><li>• Aprendizaje autónomo y colaborativo.</li><li>• Relacionar los conocimientos entre las diferentes áreas.</li><li>• Analizar y argumentar sus puntos de vista.</li><li>• Usar adecuadamente la tecnología.</li></ul>	<ul style="list-style-type: none"><li>• Honestidad.</li><li>• Honradez.</li><li>• Lealtad.</li><li>• Solidaridad.</li><li>• Responsabilidad.</li><li>• Tolerancia.</li><li>• Respeto.</li><li>• Libertad.</li></ul>





## VII. NÚMERO DE SEMESTRES PARA ACREDITAR

El programa de Estudios Técnicos Especializados en Sistemas Computacionales, Desarrollo Software quedó conformado por siete módulos temáticos a saber:

Programa de Estudios Técnicos Especializados en: <b>Sistemas Computacionales, Desarrollo de Software</b>											
PRIMER SEMESTRE											
Clave	Módulo	Modalidad	Carácter	Teóricas por semana	Prácticas por semana	Duración/semanas	Secuencia/semanas	Total de horas			Créditos
								Teóricas	Prácticas	Módulo	
	1. Análisis y Diseño de Sistemas	Curso-Taller	Obligatorio	4	2	3	1-3	12	6	18	10
	2. Estructuras Básicas de Lenguaje C	Curso-Taller	Obligatorio	4	2	4	4-7	16	8	24	10
	3. Sentencias de Control	Curso-Taller	Obligatorio	3	3	4	8-11	12	12	24	9
	4. Datos Estructurados	Curso-Taller	Obligatorio	3	3	5	12-16	15	15	30	9
<b>Subtotal</b>								<b>55</b>	<b>41</b>	<b>96</b>	<b>38</b>
SEGUNDO SEMESTRE											
	5. Funciones	Curso-Taller	Obligatorio	3	3	4	1-4	12	12	24	9
	6. Archivos	Curso-Taller	Obligatorio	3	3	6	5-10	18	18	36	9
	7. Lenguaje Visual. "Introducción a la Programación Orientada a Eventos"	Curso-Taller	Obligatorio	3	3	6	11-16	18	18	36	9
<b>Subtotal</b>								<b>48</b>	<b>48</b>	<b>96</b>	<b>27</b>
<b>Totales</b>								<b>103</b>	<b>89</b>	<b>192</b>	<b>65</b>
ACTIVIDADES PRÁCTICAS											
	Actividades Prácticas	Práctica	Obligatorio							0	0
<b>Créditos totales</b>										<b>65</b>	



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
DIRECCIÓN GENERAL DE LA ESCUELA NACIONAL COLEGIO DE CIENCIAS Y HUMANIDADES  
SECRETARÍA ACADÉMICA



DEPARTAMENTO DE OPCIONES TÉCNICAS

PROGRAMA DE ESTUDIOS TÉCNICOS ESPECIALIZADOS EN SISTEMAS COMPUTACIONALES,  
DESARROLLO DE SOFTWARE

Clave:	Semestre:	Créditos totales	Área:		Campo de conocimiento:	
	A partir de 3 <sup>er</sup> . semestre	65	Físico- Matemáticas y de las Ingenierías		Matemáticas	
Modalidad:	Carácter:	Tipo:	Seriación:	Etapa formativa:	Carga horaria a la semana	Total de semanas
Curso-Taller	Obligatorio	Teórico-Práctico	Indicativa	Propedéutica y Capacitación laboral	6	32
Módulos a cursar	Horas teóricas totales:	Horas prácticas totales:	Total de horas		Total de horas de Actividades Prácticas	Total de horas para acreditación
7	103	89	192		0	192



## VIII. ESTRUCTURA Y ORGANIZACIÓN DEL PROGRAMA DE FORMACIÓN DE TÉCNICO ESPECIALIZADO

### MÓDULO I. ANÁLISIS Y DISEÑO DE SISTEMAS

#### Introducción

La existencia de un sistema atraviesa por los procesos de análisis y diseño; en la primera etapa del estudio se determina el planteamiento del problema, a quién está dirigido el producto final y cuáles son sus necesidades, la factibilidad de poder llevarlo a cabo o no. Las posibles alternativas para solucionar un problema y el establecimiento de actividades a realizar en caso de que sea factible su desarrollo, dentro de la segunda fase, es decir durante el diseño.

Durante este módulo se lleva a cabo el formato de presentación del programa, el código fuente del mismo y se somete a pruebas para verificar los posibles errores que pueda tener.

#### Propósito

Al finalizar el módulo, el alumno:

- ‡ Aplicará los conceptos afines al análisis y diseño de sistemas empleando el ciclo de vida de un sistema informático para el desarrollo, la implementación y el mantenimiento de sistemas.

Horas teóricas por semana: 4      Horas prácticas por semana: 2      Total de créditos: 10  
 Horas teóricas por módulo: 12      Horas prácticas por módulo: 6      Total de horas: 18  
 Seriación: Ninguna

APRENDIZAJES	TEMÁTICA	ESTRATEGIAS	HORAS
El alumno: <ul style="list-style-type: none"> <li>• Describe los procedimientos a seguir para solucionar un problema.</li> <li>• Analiza las decisiones que puedan afectar el diseño de un sistema.</li> </ul>	1. Cambios al sistema razones y resistencias <ul style="list-style-type: none"> <li>1.1. Motivos del cambio</li> <li>1.2. Enfrentar el cambio</li> <li>1.3. Resistencia ante el cambio de un sistema</li> </ul> 2. Definición del problema	El profesor: <ul style="list-style-type: none"> <li>▪ Define grupos de trabajo, para asignación de roles dentro del mismo.</li> <li>▪ Expone el ciclo de vida de un sistema.</li> <li>▪ Establece las propuestas de problemas a sistematizar por parte del alumno.</li> </ul>	2 T 1 P Total 3 horas  2 T



<ul style="list-style-type: none"><li>• Comprende los factores que se deben tomar en cuenta al adquirir un nuevo equipo.</li><li>• Realiza tópicos para identificar necesidades y características que pueda contener el sistema a elaborar.</li><li>• Aplica actividades propias de la implementación y sustitución del sistema.</li></ul>	2.1. Estudio de definición del problema	<p>El alumno:</p> <ul style="list-style-type: none"><li>▪ Selecciona un problema y elabora una entrevista para obtener resultados posibles con la intención de implementar una solución automatizada.</li><li>▪ Realiza dos propuestas alternativas conforme al estudio del problema y al análisis de necesidades.</li><li>▪ Recopila información para establecer las posibles soluciones.</li><li>▪ Interpreta las propuestas planteadas y las presenta, para el análisis y observaciones de los alumnos del grupo.</li><li>▪ Aporta conclusiones que servirán para establecer la viabilidad de implementación o no de un sistema.</li><li>▪ Presenta los resultados del estudio del sistema.</li></ul>	1 P Total 3 horas
	2.2. Informe del estudio del sistema		2 T 1 P Total 3 horas
	3. Análisis del sistema		2 T 1 P Total 3 horas
	3.1. Recopilación de información		2 T 1 P Total 3 horas
	3.2. Análisis del problema		2 T 1 P Total 3 horas
	3.3. Informe del análisis		2 T 1 P Total 3 horas
4. Diseño del Sistema	2 T 1 P Total 3 horas		
4.1. Determinación de opciones	2 T 1 P Total 3 horas		
4.2. Herramientas y técnicas de diseño	2 T 1 P Total 3 horas		
4.3. Algoritmos, diagramas de flujo y pseudocódigos	2 T 1 P Total 3 horas		
4.4. Selección de una opción	2 T 1 P Total 3 horas		
4.5. Informe de diseño	2 T 1 P Total 3 horas		
5. Implementación del sistema	2 T 1 P Total 3 horas		
5.1. Consecuencias de anteriores instalaciones	2 T 1 P Total 3 horas		
5.2. Instalación del equipo	2 T 1 P Total 3 horas		
5.3. Conversión y cambio del sistema	2 T 1 P Total 3 horas		
6. Revisión posterior a la implementación	2 T 1 P Total 3 horas		



### Recursos didácticos

- Equipo de cómputo
- Software

### Evaluación

- ✓ *Diagnóstica*: Cuestionario relacionado con los tópicos del módulo: análisis, diseño, sistemas, diagramas y algoritmos.
- ✓ *Formativa*: Interpretación de los resultados de la entrevista, presentación de la propuesta de automatización conforme al estudio de viabilidad, diagramas del sistema.
- ✓ *Sumativa*: Examen correspondiente a la temática del módulo abordada.

### Mesografía básica

- 📖 Gómez, J. y Gil, F. (2010). *Administración avanzada de sistemas informáticos*. México: Alfaomega.
- 📖 Menchén, A. (2010). *Diseño de Programas*. España: Alfaomega- Ra-Ma.
- 📖 Pantaleo, G. (2011). *Calidad en el desarrollo de Software*. México: Alfaomega.
- 📖 Roger, P. (2010). *Ingeniería de Software*. (7ª Ed.). México: McGraw-Hill.

### Mesografía complementaria

- 📖 Baca, G. (2005). *Formulación y Evaluación de proyectos Informáticos*. México: McGraw-Hill.
- 📖 Gómez, A. y Suárez, C. (2012). *Sistemas de Información – Herramientas Prácticas para la gestión*. España: Alfaomega, Ra-Ma.



## MÓDULO II. ESTRUCTURAS BÁSICAS DEL LENGUAJE C

### Introducción

La realización de un análisis de sistemas facilita el entendimiento de la segmentación de un programa debido a que ambos trabajan con bloques. Iniciar con el trabajo de programación principia por conocer el entorno de trabajo del lenguaje en turno, el cual se integra por un compilador, un depurador y un editor de los comandos operacionales del mismo. Además de identificar qué es una palabra reservada, una variable, los datos, las operaciones, las librerías para poder escribir un programa y así continuar con la estructura de cualquier programa a realizar en lenguaje C, involucra una entrada de datos, el procesamiento de los mismos y la salida: resultado del procesamiento de información al interior del programa.

En este módulo se abordan las bases de la programación y sirve como plataforma para el diseño y elaboración de programas más complejos que involucren diversas posibilidades a elegir, de acuerdo con ciertos criterios de ejecución de procesos en forma finita.

### Propósito

Al finalizar el módulo, el alumno:

- ‡ Desarrollará programas estructurados haciendo uso del entorno de desarrollo del lenguaje de programación “C”, así como las estructuras básicas que le permitirán iniciarse en el diseño de aplicaciones más elaboradas.

Horas teóricas por semana: 4      Horas prácticas por semana: 2      Total de créditos: 10  
 Horas teóricas por módulo: 16      Horas prácticas por módulo: 8      Total de horas: 24  
 Seriación: Módulo I. Análisis y diseño de sistemas

APRENDIZAJES	TEMÁTICA	ESTRATEGIAS	HORAS
El alumno: <ul style="list-style-type: none"> <li>• Utiliza el entorno integrado de desarrollo de lenguaje C.</li> <li>• Aplica los conceptos fundamentales de la programación en lenguaje C.</li> </ul>	1. El entorno integrado de desarrollo <ul style="list-style-type: none"> <li>1.1. Instalación</li> <li>1.2. Arranque</li> <li>1.3. Editor</li> <li>1.4. Almacenamiento en archivo</li> <li>1.5. Compilador</li> </ul>	El profesor: <ul style="list-style-type: none"> <li>▪ Compara el entorno integrado de desarrollo de C con los entornos de paqueterías de uso común.</li> <li>▪ Resalta cada una de las etapas de un programa: la planeación, la creación de algoritmo, su</li> </ul>	4 T 0 P. Total 4 horas



<ul style="list-style-type: none"><li>• Emplea la estructura de los programas en lenguaje C.</li><li>• Diferencia constantes de variables y tipos de datos.</li><li>• Utiliza los operadores del lenguaje, así como su prioridad y orden de evaluación.</li><li>• Utiliza las funciones de entrada y salida estándar para crear programas.</li></ul>	<ol style="list-style-type: none"><li>1.6. Recuperación de archivo</li><li>1.7. Salir</li><li>2. Conceptos fundamentales<ol style="list-style-type: none"><li>2.1. Variables y Constantes</li><li>2.2. Palabras reservadas</li><li>2.3. Identificadores</li><li>2.4. Estructura básica de un programa</li></ol></li><li>3. Tipos de datos y tamaño<ol style="list-style-type: none"><li>3.1. Sistemas numéricos<ol style="list-style-type: none"><li>3.1.1. Binario</li><li>3.1.2. Hexadecimal</li></ol></li><li>3.2. Int</li><li>3.3. Float</li><li>3.4. Char</li><li>3.5. Apuntadores</li></ol></li><li>4. Declaración de datos<ol style="list-style-type: none"><li>4.1. Constantes</li><li>4.2. Variables</li><li>4.3. Enumerados</li></ol></li><li>5. Operadores<ol style="list-style-type: none"><li>5.1. Aritméticos</li><li>5.2. Relacionales</li><li>5.3. Lógicos</li><li>5.4. Conversiones</li><li>5.5. Lógicos para el manejo de bits</li><li>5.6. Asignación</li><li>5.7. Prioridad y orden de evaluación</li></ol></li></ol>	<p>codificación, la edición, el almacenamiento, la compilación, la depuración y la ejecución.</p> <ul style="list-style-type: none"><li>▪ Lista los tipos de datos que se utilizan en el lenguaje de programación C.</li><li>▪ Ejemplifica la jerarquía de operaciones</li><li>▪ Describe la estructura de un programa.</li><li>▪ Plantea ejercicios que involucren la solución de los mismos empleando las etapas de un programa.</li><li>▪ Retoma los ejercicios de creación de sistemas para su programación.</li></ul> <p>El alumno, mediante la resolución de problemas, conoce y emplea los conceptos fundamentales para la programación:</p> <ul style="list-style-type: none"><li>▪ Identifica los elementos básicos de sintaxis y semántica del lenguaje C.</li><li>▪ Identifica las diferentes secciones de un programa en C.</li><li>▪ Aplica los fundamentos de programación en la solución de problemas.</li></ul>	<p>4 T 0 P. Total 4 horas</p> <p>2 T 0 P. Total 2 horas</p> <p>2 T 2 P. Total 4 horas</p> <p>4 T 2 P. Total 6 horas</p>
--	---	---	---



	6. Rutinas de entrada y salida 6.1. Funciones de entrada estándar 6.2. Funciones de salida estándar		2 T 2 P. Total 4 horas
--	---	--	------------------------------

### Recursos didácticos

- Equipo de cómputo
- Software

### Evaluación

- ✓ *Diagnóstica*: Cuestionario de conceptos de programación.
- ✓ *Formativa*: Prácticas, código fuente de los programas elaborados, tabla de tipos de datos y operadores
- ✓ *Sumativa*: Examen de programación básica.

### Mesografía básica

- 📖 Cairo, D. (2006). *Fundamentos de programación, Piensa en C.* (12ª Ed.). México: Prentice-Hall.
- 📖 Corona, M. y Ancona, M. (2011). *Diseño de Algoritmos y su Codificación en Lenguaje C.* México: McGraw-Hill.
- 📖 Joyanes, L. y Zahonero, I. (2005). *Programación en C.* (3ª Ed.). México: McGraw-Hill.
- 📖 Méndez, A. (2013). *Diseño de Algoritmos y su programación en C.* México: Alfaomega.

### Mesografía complementaria

- 📖 Menchén, A. (2010). *Diseño de Programas.* España: Alfaomega- Ra-Ma.
- 📖 Roger, P. (2010). *Ingeniería de Software.* (7ª Ed.). México: McGraw-Hill.





## MÓDULO III. SENTENCIAS DE CONTROL

### Introducción

Previamente se ha conocido la forma de programación básica que se consigue con la entrada de datos, ejecución de procesos y el despliegue de resultados, la identificación del tipo de datos a evaluar; esto es una programación lineal, por lo que ahora abordaremos situaciones diferentes que involucren ejecuciones de una parte del código en forma finita. Esto es posible con el uso de sentencias de control.

Las sentencias se dividen en: sentencias condicionales, que dividen el flujo del programa en varias secciones dependiendo de una condición, y en sentencias cíclicas, las cuales repiten parte del código un determinado número de veces.

Bajo el principio de recursividad, las sentencias de control pueden ser utilizadas en diversas ocasiones con el uso de funciones y subprogramas.

### Propósito

Al finalizar el módulo, el alumno:

- ⚡ Simplificará y fortalecerá la elaboración de programas, utilizando sentencias de control, de selectividad y de repetición, para facilitar la lectura del código por su sencillez.

Horas teóricas por semana: 3      Horas prácticas por semana: 3      Total de créditos: 9  
 Horas teóricas por módulo: 12      Horas prácticas por módulo: 12      Total de horas: 24  
 Seriación: Módulo II. Estructuras básicas del lenguaje C

APRENDIZAJES	TEMÁTICA	ESTRATEGIAS	HORAS
El alumno: <ul style="list-style-type: none"> <li>• Identifica los distintos tipos de sentencias de selección e iteración.</li> </ul>	1. De selección 1.1. <i>If - else</i> 1.2. Switch 1.3. Sentencia <i>break</i>	El profesor: <ul style="list-style-type: none"> <li>• Demuestra las situaciones que se involucran en una toma de decisiones.</li> <li>• Señala los argumentos que integran una sentencia de</li> </ul>	6 T 6 P Total 12 horas



<ul style="list-style-type: none"><li>• Manipula las sentencias de selección e iteración para la elaboración de programas.</li><li>• Utiliza de manera adecuada cada una de las sentencias de selección e iteración.</li><li>• Conoce la importancia del uso de la sentencia <i>break</i> en las sentencias de selección e iteración.</li></ul>	<p>2. De iteración</p> <p>2.1. <i>For</i></p> <p>2.2. <i>While</i></p> <p>2.3. <i>Do</i></p>	<p>control e iteración.</p> <ul style="list-style-type: none"><li>• Ejemplifica cualquier problema que involucre sentencias de control.</li><li>• Propone ejercicios para la resolución por parte de los alumnos.</li><li>• Menciona las alternativas del proyecto a realizar correspondiente a este módulo.</li></ul> <p>El alumno:</p> <ul style="list-style-type: none"><li>• Realiza una investigación acerca de las instrucciones de selección y de iteración.</li><li>• Integra un equipo y elige el proyecto del módulo a desarrollar (calculadora, cajero, nota de venta).</li><li>• Resuelve los ejercicios propuestos por el profesor utilizando el lenguaje de programación C.</li><li>• Elabora los diagramas de flujo conforme al proyecto seleccionado por el equipo.</li><li>• Elabora el manual técnico y de usuario del proyecto.</li></ul>	<p>6 T 6 P Total 12 horas</p>
---	--	--	---------------------------------------

### Recursos didácticos

- Equipo de cómputo
- Software



## Evaluación

- ✓ *Diagnóstica*: Planteamiento, desarrollo y propuesta de solución a cualquier problema que involucre la repetición de instrucciones u operaciones para identificar formas de solucionarlo.
- ✓ *Formativa*: Prácticas y programas realizados que involucren sentencias de control.
- ✓ *Sumativa*: Examen de programación que involucre tópicos relacionados con sentencias selectivas y de iteración.

## Mesografía básica

- 📖 Gottfried, B. (2005). *Programación en C*. (2ª Ed.). México: McGraw-Hill.
- 📖 Joyanes, L. y Zahonero, I. (2005). *Programación en C*. (3ª Ed.). México: McGraw-Hill.
- 📖 Villalobos, R. (2014). *Fundamentos de programación C++ -Más de 100 algoritmos codificados-*. (2ª Ed.). México: Alfaomega; Macro.

## Mesografía complementaria

- 📖 King, K. (2008). *C Programming: A Modern Approach*. (2ª Ed.). Estados Unidos: WWNorton.
- 📖 Menchén, A. (2010). *Diseño de Programas*. España: Alfaomega- Ra-Mal.
- 📖 Roger, P. (2010). *Ingeniería de Software*. (7ª Ed.). México: McGraw-Hill.
- 📖 Van Der, P. (2007). *Expert C Programming Deep C Secrets*. Prentice Hall.



## MÓDULO IV. DATOS ESTRUCTURADOS

### Introducción

El establecer una metodología de programación estructurada permite la segmentación de procesos por medio de una identificación de tareas bien definidas, facilitando la lectura dentro del programa e incrementando el rendimiento del mismo. Esta metodología de programación se caracteriza por hacer uso de las estructuras de control: iteración, selección y control trabajadas anteriormente.

Los datos estructurados implican la generación de datos compuestos que, a diferencia de lo estudiado en los módulos anteriores, son variables que pueden almacenar varios valores de manera simultánea; éstos pueden ser registros, arreglos y las mismas estructuras. La forma para poder acceder a los valores almacenados se conoce como ordenamiento y búsqueda; las cuales permiten ubicar datos en determinada posición y realizar operaciones como: consulta, actualización de valores y eliminación de información almacenada.

El manejo de estructuras sienta las bases del manejo de información en archivos cuyo propósito es el almacenamiento de información.

### Propósito

Al finalizar el módulo, el alumno:

- Generará diferentes tipos de datos compuestos para optimizar el manejo de datos y aplicará los métodos de ordenamiento y búsqueda.

Horas teóricas por semana: 3      Horas prácticas por semana: 3      Total de créditos: 9  
 Horas teóricas por módulo: 15      Horas prácticas por módulo: 15      Total de horas: 30  
 Seriación: Módulo III. Sentencias de control

APRENDIZAJES	TEMÁTICA	ESTRATEGIAS	HORAS
El alumno: <ul style="list-style-type: none"> <li>• Elabora programas basados en arreglos para almacenamiento temporal de información, disponible al</li> </ul>	1. Arreglos 1.1. Cadenas de caracteres 1.2. Matrices 1.3. Arreglos multidimensionales	El profesor: <ul style="list-style-type: none"> <li>• Aclara los conceptos de cadena, vectores y matrices, y muestra cómo son empleados en la programación.</li> </ul>	5 T 5 P Total 10 horas



<p>momento de la ejecución del mismo.</p> <ul style="list-style-type: none"><li>• Diferencia el uso de arreglos vectoriales y arreglos matriciales.</li><li>• Interpreta programas basados en algoritmos de ordenamiento y búsqueda.</li><li>• Genera estructuras de almacenamiento a través de arreglos.</li><li>• Diferencia entre las instrucciones de los métodos de búsqueda y de ordenamiento.</li></ul>	<p>2. Estructuras 2.1. Declaración 2.2. Manipulación 2.3. Clases</p> <p>3. Ordenamiento y búsqueda 3.1. Método de la burbuja 3.2. Métodos de búsqueda secuencial y binaria</p>	<ul style="list-style-type: none"><li>• Muestra el empleo de los arreglos vectoriales y matriciales, haciendo uso de problemas que satisfagan estos principios.</li><li>• Propone temas de investigación conforme a los métodos de ordenamiento y búsqueda básicos.</li><li>• Propone ejercicios que abunden en la temática del módulo para que el alumno los resuelva.</li></ul> <p>El alumno:</p> <ul style="list-style-type: none"><li>• Investiga los algoritmos de búsqueda y de ordenación existentes.</li><li>• Organizado en equipos no mayores a 3 integrantes, selecciona un método de búsqueda y explica su funcionamiento, uso y características que lo identifican.</li><li>• Elabora una agenda o directorio con la ayuda de arreglos y estructuras.</li><li>• Realiza las adecuaciones pertinentes al proyecto seleccionado para implementar algún método de búsqueda con el fin de encontrar los datos y la ubicación de un registro.</li></ul>	<p>5 T 5 P Total 10 horas</p> <p>5 T 5 P Total 10 horas</p>
--	--	---	---



- |  |  |   |  |
|--|--|---|--|
|  |  | <ul style="list-style-type: none"><li>• Incorpora uno de los métodos de búsqueda para agregar o eliminar la información contenida en el proyecto.</li></ul> |  |
|--|--|---|--|

### Recursos didácticos

- Equipo de cómputo
- Software

### Evaluación

- ✓ *Diagnóstica*: Planteamiento, desarrollo y propuesta de solución a cualquier problema que involucre la repetición de instrucciones u operaciones para identificar formas de solucionarlo.
- ✓ *Formativa*: Prácticas y programas realizados que involucren sentencias de control.
- ✓ *Sumativa*: Investigación de los métodos de búsqueda u ordenamiento, exposición del método de ordenamiento o búsqueda, programa del directorio o agenda y programa del directorio o agenda con las funciones de búsqueda y ordenamiento incorporadas.

### Mesografía básica

- 📖 Gottfried, B. (2005). *Programación en C*. (2ª Ed.). México: McGraw-Hill.
- 📖 Joyanes, L. y Zahonero, I. (2005). *Programación en C*. (3ª Ed.). México: McGraw-Hill.
- 📖 Villalobos, R. (2014). *Fundamentos de programación C++ -Más de 100 algoritmos codificados-*. (2ª Ed). México: Alfaomega; Macro.

### Mesografía complementaria

- 📖 King, K. (2008), *C Programming: A Modern Approach*. (2ª Ed.). Estados Unidos: WWNorton.
- 📖 Menchén, A. (2010). *Diseño de Programas*. España: Alfaomega- Ra-Mal.
- 📖 Roger, P. (2010). *Ingeniería de Software*. (7ª Ed.). México: McGraw-Hill.
- 📖 Van Der, P. (2007). *Expert C Programming Deep C Secrets*. Prentice Hall.



## MÓDULO V. FUNCIONES

### Introducción

Hasta ahora la codificación en el lenguaje C ha abordado las estructuras secuenciales, lo que proporciona soluciones a programas lineales, es decir, comienzan por la primera instrucción y acaban por la última, ejecutándose todas una sola vez. Las funciones son una técnica de la programación estructurada que permite dividir en módulos un programa, permitiendo el almacenamiento, el intercambio y el cálculo de valores a través del manejo de variables globales y locales e incluso de registros, complementando el trabajo con la incorporación de funciones gráficas para la interpretación visual de resultados cuya programación así lo requiera.

Las funciones son el inicio para el manejo de datos estructurados donde se dividen los datos pero se almacenan de forma independiente y conjunta al mismo tiempo.

### Propósitos

Al finalizar el módulo, el alumno:

- ⚡ Aplicará concepto y técnicas de modularidad en la elaboración de programas extensos y complejos, empleando procedimientos y funciones para facilitar el diseño, análisis e interpretación de sistemas.
- ⚡ Empleará el modo gráfico dentro de los programas, utilizando las funciones básicas del entorno de programación C para mejorar la presentación de datos e información.

Horas teóricas por semana: 3      Horas prácticas por semana: 3      Total de créditos: 9  
 Horas teóricas por módulo: 12      Horas prácticas por módulo: 12      Total de horas: 24  
 Seriación: Módulo IV. Datos estructurados

APRENDIZAJES	TEMÁTICA	ESTRATEGIAS	HORAS
El alumno: <ul style="list-style-type: none"> <li>• Describe la importancia de estructurar programas en forma modular.</li> <li>• Desarrolla programas utilizando el concepto de</li> </ul>	1. Modularidad 1.1. Concepto 1.2. Importancia	El profesor: <ul style="list-style-type: none"> <li>▪ Define los principios de modularidad.</li> <li>▪ Demuestra los tipos de modularidad que existen y pueden emplearse en el desarrollo de un programa.</li> </ul>	2 T 0 P Total 2 horas



<p>modularidad por medio de las funciones.</p> <ul style="list-style-type: none"><li>• Utiliza el paso de parámetros por valor o referencia dentro y fuera de las funciones.</li><li>• Utiliza las funciones predefinidas en las librerías de C.</li><li>• Identifica la diferencia de utilizar variables externas, estáticas y de registro.</li><li>• Distingue entre los distintos modos de video en los que trabaja C y la inicialización de cada uno.</li><li>• Realiza programas que corran en el modo gráfico de C.</li></ul>	<p>2. Funciones 2.1. Definición 2.1.1. Cuerpo de la función 2.1.2. Valor retornado por una función 2.2. Llamada a una función 2.3. Declaración de una función 2.4. Paso de parámetros</p> <p>3. Librerías 3.1. Definición 3.2. Funciones estándar</p> <p>4. Variables 4.1. Externas 4.2. Estáticas 4.3. Registro</p> <p>5. Funciones Gráficas 5.1. Modos de video 5.2. Inicialización 5.3. Primitivas</p>	<ul style="list-style-type: none"><li>▪ Ejemplifica los diversos tipos de modularidad a través de ejercicios básicos que permitan hacer uso de este principio.</li><li>▪ Propone ejercicios para la resolución por parte de los alumnos.</li><li>▪ Realiza ejercicios empleando el modo gráfico propio del lenguaje de programación C.</li><li>▪ Propone ejercicios que involucren el manejo de la interfaz gráfica.</li></ul> <p>El alumno:</p> <ul style="list-style-type: none"><li>▪ Resuelve problemas que propicien la necesidad de utilizar el concepto de programación modular como una estrategia para la resolución de problemas complejos o extensos.</li><li>▪ Utiliza las librerías, funciones estándar y diferentes tipos de variables que contienen las definiciones y declaraciones que se emplean en todo programa.</li><li>▪ Elabora programas utilizando el modo gráfico y sus principales comandos (primitivas), para la elaboración de desplegados gráficos.</li></ul>	<p>2 T 3 P Total 5 horas</p> <p>2 T 3 P Total 5 horas</p> <p>3 T 3 P Total 6 horas</p> <p>3 T 3 P Total 6 horas</p>
---	---	---	---





### Recursos didácticos

- Equipo de cómputo
- Software

### Evaluación

- ✓ *Diagnóstica*: No se realiza.
- ✓ *Formativa*: Prácticas del módulo.
- ✓ *Sumativa*: Proyecto del módulo anterior utilizando modo gráfico y principios de modularidad.

### Mesografía básica

- 📖 Gottfried, B. (2005). *Programación en C*. (2ª Ed.). México: McGraw-Hill.
- 📖 Joyanes, L. y Zahonero, I. (2005). *Programación en C*. (3ª Ed.). México: McGraw-Hill.
- 📖 Villalobos, R. (2014). *Fundamentos de programación C++ -Más de 100 algoritmos codificados-*. (2ª Ed.). México: Alfaomega; Macro.

### Mesografía complementaria

- 📖 King, K. (2008). *C Programming: A Modern Approach*. (2ª Ed.). Estados Unidos: WWNorton.
- 📖 Menchén, A. (2010). *Diseño de Programas*. España: Alfaomega- Ra-Ma.
- 📖 Roger, P. (2010). *Ingeniería de Software*. (7ª Ed.). México: McGraw-Hill.
- 📖 Van Der, P. (2007). *Expert C Programming Deep C Secrets*. Prentice Hall.



## MÓDULO VI. ARCHIVOS

### Introducción

El mundo actual necesita almacenar los diversos datos con los que trabajan los sistemas, en archivos fáciles de transportar, para poder trabajar con ellos sin necesidad de editarlos nuevamente al ejecutar el sistema, por lo que es conveniente poder leer, escribir, modificar y borrar los datos con los que trabajan los sistemas desde el sistema mismo.

### Propósito

Al finalizar el módulo, el alumno:

- ⌘ Manejará los mecanismos de creación y acceso a archivos de tipo texto y binario para almacenar información de manera secuencial o aleatoria dentro de archivos o estructuras ampliando las posibilidades de un programa o sistema.

Horas teóricas por semana: 3      Horas prácticas por semana: 3      Total de créditos: 9  
 Horas teóricas por módulo: 18      Horas prácticas por módulo: 18      Total de horas: 36  
 Seriación: Módulo V. Funciones

APRENDIZAJES	TEMÁTICA	ESTRATEGIAS	HORAS
El alumno: <ul style="list-style-type: none"> <li>• Diferencia los archivos de tipo texto y los archivos binarios.</li> <li>• Emplea funciones de lectura y escritura de datos en archivos.</li> <li>• Aplica los conceptos de acceso secuencial y aleatorio para la manipulación del contenido del archivo.</li> </ul>	1. Archivos <ul style="list-style-type: none"> <li>1.1. Tipo                             <ul style="list-style-type: none"> <li>1.1.1. Modo Texto</li> <li>1.1.2. Modo Binario</li> </ul> </li> <li>1.2. Apertura                             <ul style="list-style-type: none"> <li>1.2.1. fopen()</li> </ul> </li> <li>1.3. Cierre                             <ul style="list-style-type: none"> <li>1.3.1. fclose()</li> </ul> </li> </ul> 2. Escritura y Lectura <ul style="list-style-type: none"> <li>2.1. Entrada y salida formateado                             <ul style="list-style-type: none"> <li>2.1.1. fscanf()</li> </ul> </li> <li>2.2. Entrada y salida de carácter                             <ul style="list-style-type: none"> <li>2.2.1. fgetc()</li> </ul> </li> </ul>	El profesor: <ul style="list-style-type: none"> <li>• Muestra las funciones de entrada y salida, de tipo carácter, directa y formateada.</li> <li>• Ejemplifica los tipos de archivos que existen en programación C, la forma en que se ingresa a su contenido, cómo se modifica y cómo se guardan los cambios.</li> <li>• Formula una idea general a partir de una lluvia de ideas sobre lo que es un archivo, para qué sirve y en qué se utiliza.</li> </ul>	6 T 6 P Total 12 horas  6 T 6 P Total 12 horas



<ul style="list-style-type: none"><li>• Soluciona problemas o errores generados por emplear archivos.</li><li>• Realiza programas que involucren las actividades propias del manejo de archivos.</li></ul>	<ul style="list-style-type: none"><li>2.2.2. fgets()</li><li>2.2.3. fputc()</li><li>2.2.4. fputs()</li><li>2.3. Entrada y salida directa<ul style="list-style-type: none"><li>2.3.1. fread()</li><li>2.3.2. fwrite()</li></ul></li><li>2.4. Detección del fin<ul style="list-style-type: none"><li>2.4.1. feof()</li></ul></li><li>3. Acceso secuencia y aleatorio<ul style="list-style-type: none"><li>3.1. Funciones secuenciales<ul style="list-style-type: none"><li>3.1.1. ftell()</li><li>3.1.2. rewind()</li></ul></li><li>3.2. Función aleatoria<ul style="list-style-type: none"><li>3.2.1. fseek()</li></ul></li></ul></li></ul>	<p>El alumno:</p> <ul style="list-style-type: none"><li>• Selecciona un posible proyecto a elaborar de una lista generada a partir de la lluvia de ideas.</li><li>• Investiga los principales tipos de formatos de almacenamiento de datos en un archivo.</li><li>• Elabora el programa correspondiente al proyecto seleccionado, mismo que incluirá:<ul style="list-style-type: none"><li>- Funciones de apertura y cierre de un archivo.</li><li>- Búsqueda de registros al interior del archivo.</li><li>- Funciones que permitan llevar a cabo las ordenes de guardar y sobrescribir datos en un archivo.</li></ul></li><li>• Realiza una tabla donde se establecen características y diferencias entre archivos de texto y archivos binarios.</li></ul>	<p>6 T 6 P Total 12 horas</p>
--	--	--	---------------------------------------

### Recursos didácticos




- Equipo de cómputo
- Software







## Evaluación

- ✓ *Diagnóstica*: Cuestionario relacionado con los conceptos de archivos, estructuras, datos y formas de almacenamiento.
- ✓ *Formativa*: Explicación del código fuente del programa entregado, uso de funciones para el manejo de archivos.
- ✓ *Sumativa*: El programa general de este módulo (agenda o directorio).

## Mesografía básica

-  Gottfried, B. (2005). *Programación en C*. (2ª Ed.). México: McGraw-Hill.
-  Joyanes, L. y Zahonero, I. (2005). *Programación en C*. (3ª Ed.). México: McGraw-Hill.
-  Villalobos, R. (2014). *Fundamentos de programación C++ -Más de 100 algoritmos codificados-*. (2ª Ed.). México: Alfaomega; Macro.

## Mesografía complementaria

-  King, K. (2008). *C Programming: A Modern Approach*. (2ª Ed.). Estados Unidos: WWNorton.
-  Menchén, A. (2010). *Diseño de Programas*. España: Alfaomega- Ra-Mal.
-  Roger, P. (2010). *Ingeniería de Software*. (7ª Ed.). México: McGraw-Hill.
-  Van Der, P. (2007). *Expert C Programming Deep C Secrets*. Prentice Hall.



## MÓDULO VII. LENGUAJE VISUAL “INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A EVENTOS”

### Introducción

Familiarizarse con el entorno de trabajo, como se realizó con el Lenguaje C simplificado el diseño y desarrollo de aplicaciones. Partir de la estructura básica de un programa permite alcanzar la comprensión del modelo estructurado. Aplicando esta misma metodología abordaremos la programación Orientada a Eventos, incorporando las estructuras de programación a las acciones definidas para un determinado objeto existente en nuestra interfaz gráfica.

Utilizar el lenguaje de programación Visual Basic, orientado a eventos por la sencillez de su entorno, resulta simple para la incorporación de programación estructurada dentro de los módulos asociados a eventos de un objeto.

### Propósito

Al finalizar el módulo, el alumno:

- Conocerá el ambiente de programación orientada a eventos, desarrollará aplicaciones básicas empleado los principios y las bases de la programación estructurada en el diseño modular, para determinar la migración de cualquier aplicación desarrollada.

Horas teóricas por semana: 3      Horas prácticas por semana: 3      Total de créditos: 9  
 Horas teóricas por módulo: 18      Horas prácticas por módulo: 18      Total de horas: 36  
 Seriación: Módulo VI. Archivos

APRENDIZAJES	TEMÁTICA	ESTRATEGIAS	HORAS
El alumno: <ul style="list-style-type: none"> <li>• Obtiene un panorama general de la organización de un proyecto en cualquier entorno propio de los lenguajes visuales.</li> <li>• Relaciona cada uno de los objetos con los eventos correspondientes.</li> </ul>	1. El entorno de Visual Basic <ul style="list-style-type: none"> <li>1.1. Características de Visual Basic</li> <li>1.2. Pantalla Principal                             <ul style="list-style-type: none"> <li>1.2.1. Barra de Títulos</li> <li>1.2.2. Barra de Menús</li> <li>1.2.3. Barra de Herramientas</li> </ul> </li> <li>1.3. Ventana del Proyecto</li> <li>1.4. Menú de ayuda</li> </ul>	El profesor: <ul style="list-style-type: none"> <li>• Mediante una exploración dirigida demuestra el entorno de Visual Basic, los elementos que lo integran, las herramientas y los menús que lo integran.</li> </ul> El alumno: <ul style="list-style-type: none"> <li>• Identifica los elementos básicos de sintaxis y</li> </ul>	6 T 6 P Total 12 horas



<ul style="list-style-type: none"><li>• Identifica propiedades exclusivas de acuerdo con la función del objeto en turno.</li><li>• Soluciona problemas a través de elementos propios del lenguaje de programación diseñando una interfaz legible que muestre resultados de manera sencilla.</li></ul>	<ol style="list-style-type: none"><li>2. Herramientas de programación y su funcionamiento<ol style="list-style-type: none"><li>2.1. La caja de herramientas</li><li>2.2. Creación de controles<ol style="list-style-type: none"><li>2.2.1. Propiedades del control</li><li>2.2.2. Procedimientos del control</li></ol></li></ol></li><li>3. Programación de Eventos<ol style="list-style-type: none"><li>3.1. Diseño de la interfaz</li><li>3.2. Definición de variables y constantes</li><li>3.3. Asignación de propiedades de un evento</li><li>3.4. Control de flujo<ol style="list-style-type: none"><li>3.4.1. Sentencias de programación</li><li>3.4.2. Funciones</li></ol></li><li>3.5. Ejecución y depuración del código</li><li>3.6. Depuración del formulario</li></ol></li></ol>	<p>semántica de Visual Basic.</p> <ul style="list-style-type: none"><li>• Identifica las diferentes zonas de un programa de Visual Basic.</li></ul> <p>El profesor:</p> <ul style="list-style-type: none"><li>• Mediante la resolución de problemas, encuadra a los alumnos los conceptos de Visual Basic y su relación con la programación orientada a eventos.</li><li>• Retoma algunos ejercicios para realizar programas en Visual Basic.</li></ul> <p>El alumno:</p> <ul style="list-style-type: none"><li>• Resuelve ejercicios propuestos adecuándolos a la programación orientada a eventos.</li><li>• Migra programas del tipo estructurado al entorno de programación orientada a eventos.</li></ul>	<p>6 T 6 P Total 12 horas</p> <p>6 T 6 P Total 12 horas</p>
---	---	--	---

### Recursos didácticos

- Equipo de cómputo
- Software



### Evaluación

- ✓ *Diagnóstica*: Cuestionario que aborde cuestiones de programación estructurada para conocer nivel de dominio y conocimiento del tema.
- ✓ *Formativa*: Programas.
- ✓ *Sumativa*: Entrega de un sistema de punto de venta, documentación y exposición del proyecto.

### Mesografía básica

- 📖 Carrasco, D. (2009). *Visual Basic 6.0*. México: Alfaomega-Macro.
- 📖 Ceballos, F. (2010). *Enciclopedia de Microsoft Visual Basic*. España: Alfaomega-Ra-Ma
- 📖 Ceballos, F. (2014). *Enciclopedia de Microsoft Visual Basic – Interfaces gráficas y aplicaciones para Internet con Windows Forms y ASP.NET*. España: Alfaomega-Ra-Ma.
- 📖 Villalobos, R. (2014). *Fundamentos de programación Visual Basic. Más de 100 algoritmos codificados*. México: Alfaomega-Macro.

### Mesografía complementaria

- 📖 Sánchez, C. (2008). *Desarrollando Aplicaciones con Visual Basic .NET 2008*. España: Alfaomega-Ra-Ma.



## **IX. EVALUACIÓN FINAL**

La evaluación final consistirá en la revisión del producto final mismo que puede ser presentado de manera individual o en equipo de hasta tres personas. Dicho producto es un sistema informático que sirva para dar una solución automatizada a un problema en particular, además de la documentación derivada del mismo.

## **X. ACTIVIDADES PRÁCTICAS**

Debido a las características didáctico-pedagógicas en las que se desarrolla el programa académico en el Aula-taller los alumnos en lugar de realizar actividades prácticas en alguna Institución pública o privada llevan a cabo un “Trabajo Práctico Final” en el cual tendrán que aplicar los conocimientos y las habilidades adquiridas en el curso. Para la obtención del diploma de técnico en Sistemas Computacionales, Desarrollo de Software, es necesario aprobar ambas fases del programa y presentar un trabajo el cual es el equivalente académico de las Actividades Prácticas.

Sin embargo, es trabajo pendiente del Departamento la incorporación de Actividades Prácticas a este ETE.





## XI. REQUISITOS DE INGRESO, PERMANENCIA Y EGRESO

INGRESO	PERMANENCIA	EGRESO
<ul style="list-style-type: none"><li>▪ El alumno deberá haber sido aceptado en el CCH, de acuerdo con los mecanismos institucionales establecidos para tal fin.</li><li>▪ Haber cursado el segundo semestre del Bachillerato en el CCH.</li><li>▪ Registrar su inscripción a los Estudios Técnicos Especializados por el medio que corresponda.</li><li>▪ Cumplir con los requisitos establecidos por el Departamento de Opciones Técnicas para la formalización de la inscripción.</li></ul>	<ul style="list-style-type: none"><li>▪ Haber acreditado el primer semestre.</li><li>▪ Cumplir con un mínimo de 80% de asistencia.</li><li>▪ Para el caso de las Actividades Prácticas establecidas en cada programa es necesario que el alumno cubra el total de horas en un plazo máximo de un año.</li></ul>	<ul style="list-style-type: none"><li>▪ Haber cubierto el 100% de los créditos establecidos en el programa de Estudios Técnicos Especializados.</li><li>▪ No adeudar materiales o reparar los daños que pudo haber ocasionado.</li></ul>

## XII. PERFIL PROFESIOGRÁFICO<sup>2</sup>

- ⌘ Actuaría (Áreas de Cómputo e Informática)
- ⌘ Ciencias de la Computación
- ⌘ Ciencias de la Informática
- ⌘ Informática
- ⌘ Ingeniería en Computación
- ⌘ Ingeniería en Comunicación y Electrónica
- ⌘ Ingeniería en Informática
- ⌘ Ingeniería en Telecomunicaciones

<sup>2</sup> Información de acuerdo con el documento "Perfiles Profesiográficos con propósitos de cobertura de grupos vacantes y concursos de definitividad de las áreas y departamentos académicos del Colegio de Ciencias y Humanidades", publicado en el suplemento especial de *Gaceta CCH*, Número 6, del 9 de junio de 2008.



- ⌘ Ingeniería Mecatrónica
- ⌘ Matemáticas Aplicadas y Computación

*Nota: Los aspectos generales de perfil profesiográfico de los Estudios Técnicos Especializados se encuentran explicados de manera detallada en el Plan de Estudios, pág.57.*



**COLEGIO DE CIENCIAS Y HUMANIDADES**



**Dr. BENJAMÍN BARAJAS SÁNCHEZ**  
Director General

**Dra. María Leticia de Anda Munguía**  
Secretaria General

**Lic. María Elena Juárez Sánchez**  
Secretaria Académica

**M. en A. Isaí Korina Ramos Bernal**  
Jefe del Departamento de Opciones Técnicas

**MAC. González Sánchez Diego**  
Secretario Auxiliar Académico de Opciones Técnicas